

Exponát do Aurélia
Detekcia žmurkacích gest

Dokumentácia

Ing. Peter Varga

Miroslav Baluch, Matej Dráb, Ivana Nemsilajová, Martina Veselá

Obsah

<u>Katalóg požiadaviek</u>	<u>5</u>
<u>1 Úvod</u>	<u>6</u>
<u>1.1 Účel dokumentu</u>	<u>6</u>
<u>1.2 Rozsah systému</u>	<u>6</u>
<u>1.3 Slovník pojmov</u>	<u>6</u>
<u>1.4 Referencie</u>	<u>6</u>
<u>1.5 Prehľad zvyšku dokumentu</u>	<u>7</u>
<u>2 Popis plánovaného systému</u>	<u>8</u>
<u>2.1 Perspektíva produktu</u>	<u>8</u>
<u>2.2 Funkcie produktu</u>	<u>8</u>
<u>2.3 Charakteristika používateľov</u>	<u>8</u>
<u>2.4 Všeobecné obmedzenia</u>	<u>8</u>
<u>2.5 Predpoklady a závislosti</u>	<u>9</u>
<u>3 Špecifikácia požiadaviek</u>	<u>10</u>
<u>3.1 Funkčné požiadavky</u>	<u>10</u>
<u>3.1.1 Region of interest</u>	<u>10</u>
<u>3.1.2 Výber tváre</u>	<u>10</u>
<u>3.1.3 Zobrazenie obrazu z webovej kamery v reálnom čase</u>	<u>10</u>
<u>3.1.4 Grafické zvýraznenie detekovaného objektu</u>	<u>10</u>
<u>3.1.5 Detekcia gest</u>	<u>11</u>
<u>3.1.6 Výpis po detekcii gesta</u>	<u>11</u>
<u>3.1.7 Nápona gest</u>	<u>11</u>
<u>3.1.8 Konfigurácia gest</u>	<u>12</u>
<u>3.1.9 Ukončenie gesta</u>	<u>12</u>
<u>3.1.10 Ochrana displeja (screen saver)</u>	<u>12</u>
<u>3.1.11 Varovanie pri nízkom osvetlení</u>	<u>12</u>
<u>3.2 Požiadavky rozhrania</u>	<u>13</u>
<u>3.2.1 Hardvérové rozhrania</u>	<u>13</u>
<u>3.2.2 Možnosť vybrať kameru zo zoznamu kamier</u>	<u>13</u>
<u>3.2.3 Softvérové rozhrania</u>	<u>13</u>
<u>3.2.4 Užívateľské rozhrania</u>	<u>13</u>
<u>3.3 Požiadavky, ktoré sa nevzťahujú na funkcionality</u>	<u>13</u>
<u>3.3.1 Implementačné požiadavky</u>	<u>13</u>
<u>3.3.2 Stručná dokumentácia</u>	<u>13</u>

<u>Návrh</u>	<u>14</u>
<u>1 Úvod</u>	<u>15</u>
<u>1.1 Účel dokumentu</u>	<u>15</u>
<u>2 Dekompozícia a popis</u>	<u>16</u>
<u>2.1 Python</u>	<u>16</u>
<u>2.1.1 OpenCV</u>	<u>16</u>
<u>2.1.1.1 OpenCV Funkcie</u>	<u>17</u>
<u>2.1.2 NumPy</u>	<u>17</u>
<u>2.1.2.1 NumPy Funkcie</u>	<u>18</u>
<u>2.1.3 GazeTracking</u>	<u>18</u>
<u>2.2 CMake</u>	<u>18</u>
<u>3 Návrh používateľského rozhrania</u>	<u>19</u>
<u>3.1 UI</u>	<u>19</u>
<u>4 Návrh implementácie</u>	<u>20</u>
<u>4.1 UML - Use-case diagram</u>	<u>20</u>
<u>4.2 UML - Stavový diagram</u>	<u>20</u>
<u>4.3 UML - Diagram komponentov</u>	<u>21</u>
<u>4.4 UML - Triedny diagram</u>	<u>22</u>
<u>4.5 Problém kalibrácie</u>	<u>23</u>
<u>5 Špecifikácia vonkajších interfejsov</u>	<u>24</u>
<u>6 Formáty súborov</u>	<u>25</u>
<u>6.1 Konfiguračný súbor</u>	<u>25</u>
<u>7 rozdelenie práce</u>	<u>26</u>
<u>Testovacie scenáre</u>	<u>27</u>
<u>1 Oblasť záujmu</u>	<u>28</u>
<u>1.1 Návštevník je príd'aleko / nie je pri exponáte</u>	<u>28</u>
<u>1.2 Návštevník nie je v zornom poli detekcie</u>	<u>28</u>
<u>1.3 Návštevníkova tvár je na vhodnom mieste</u>	<u>28</u>

<u>2 Zobrazenie obrazu</u>	<u>28</u>
<u>2.1 Kamera sníma</u>	<u>28</u>
<u>2.2 Širokoúhlý monitor ale nie kamera</u>	<u>28</u>
<u>2.3 Širokouhlá kamera ale nie monitor</u>	<u>29</u>
<u>2.4 Nedostatok svetla pri kamere</u>	<u>29</u>
<u>3 Zvýraznenie detekovaného objektu a ich počet</u>	<u>29</u>
<u>3.1 Detekovaná tvár</u>	<u>29</u>
<u>3.2 Viacero Tvárí</u>	<u>29</u>
<u>4 Detekcia gest</u>	<u>29</u>
<u>4.1 Správne vykonanie gesta</u>	<u>29</u>
<u>4.2 Zlé vykonanie gesta</u>	<u>30</u>
<u>4.3 Prerušenie vykonania gesta iným objektom</u>	<u>30</u>
<u>5 Konfigurácia a zapnutie</u>	<u>30</u>
<u>5.1 Zmena nastavení</u>	<u>30</u>
<u>5.2 Nevhodný konfiguračný súbor</u>	<u>30</u>
<u>6 Ochrana displeja</u>	<u>30</u>
<u>6.1 Nič sa nedeje</u>	<u>30</u>
<u>6.2 K exponátu sa priblíži návštevník</u>	<u>31</u>
<u>6.3 Kamera neustále sníma návštevníka</u>	<u>31</u>
<u>7 Kamera</u>	<u>31</u>
<u>7.1 Viacero kamier</u>	<u>31</u>
<u>7.2 Žiadna kamera</u>	<u>31</u>
<u>8 Detekcia úkonov</u>	<u>31</u>
<u>8.1 Zatvorenie oboch očí</u>	<u>31</u>
<u>8.2 Zatvorenie ľavého oka</u>	<u>31</u>
<u>8.3 Zatvorenie pravého oka</u>	<u>32</u>
<u>8.4 Pohľad doprava</u>	<u>32</u>
<u>8.5 Pohľad doľava</u>	<u>32</u>
<u>8.6 Úsmev</u>	<u>32</u>
<u>8.7 Vyplazený jazyk</u>	<u>32</u>
<u>Návod</u>	<u>32</u>

Katalóg požiadaviek

1 Úvod

1.1 Účel dokumentu

Tento dokument slúži ako návrh na schválenie pre zadávateľa a ako referencia pre vývojový tím.

Katalóg popisuje požiadavky zadávateľa na funkcionality softvéru na detekciu žmurkacích gest.

Predmetom dokumentu je špecifikovať požiadavky finálneho produktu. Špecifikácia bude slúžiť ako východisko pre vyhodnocovanie správnosti softvéru. Katalóg je určený pre všetkých stakeholderov, čiže pre zadávateľa, vývojový tím a koordinátora projektu.

1.2 Rozsah systému

Hlavným cieľom je vytvoriť softvér, ktorý bude za použitia webkamery detekovať tvár a oči používateľa. Systém bude detekovať rôzne druhy žmurkacích kombinácií, čo budú gestá na základe, ktorých sa používateľovi vypíše odpoveď. Softvér bude umiestnený v Zážitkovom centre vedy - Aurelium.

1.3 Slovník pojmov

OpenCV - softvérová knižnica určená pre počítačové videnie a strojové učenie

Region of Interest - oblasť záujmu (časť obrazu, na ktorú sa chceme zamerať)

Aspect ratio - pomer strán obrazu

fps - obrazová frekvencia, počet obrazov zobrazených, zaznamenaných alebo prenesených za sekundu

fullscreen - zobrazenie na celú obrazovku

úkon - detekovaná mimika tváre ako žmurknutie, zavretie očí atď.

gesto - postupnosť úkonov

1.4 Referencie

[Dokumentácia k OpenCV](#)

[Tutoriál k detekcii tváre](#)

1.5 Prehľad zvyšku dokumentu

Ďalšia kapitola dokumentu popisuje perspektívu systému, jeho funkcie a charakterizuje používateľa, pre ktorého je softvér určený. Tretia technická kapitola obsahuje kompletný zoznam funkčných a kvalitatívnych požiadaviek.

2 Popis plánovaného systému

2.1 Perspektíva produktu

Nasledovný systém bude využívaný v zážitkovom centre vedy Aurelium v Bratislave. Jeho hlavnou úlohou je vedieť detekovať tváre okolitých ľudí a vedieť zobrazit' obrazce podľa zadanej sekvencie úkonov (pokiaľ je táto sekvencia úkonov definovaná v nastaveniach). Systém bude možné použiť na ktorejkoľvek webkamere, ktorá je kompatibilná so systémom Windows.

2.2 Funkcie produktu

Systém bude pomocou webkamery schopný nájsť tvár používateľa a následne detekovať úkony a gestá. Sleduje oči používateľa, vie detekovať či je oko otvorené alebo zavreté a ako dlho je zavreté. Návštevník exponátu takto vykoná vopred nastavené gesto kombináciou úkonov, čiže napríklad žmurkania pravého alebo ľavého oka a ich dĺžkou. Po detekcii gesta systém oznámi návštevníkovi, ktoré gesto bolo vykonané.

2.3 Charakteristika používateľov

Systém je určený pre návštevníkov centra vedy Aurelium všetkých vekových kategórií. Bežný návštevník výstavy bude mať možnosť vyskúšať si detekciu gest podľa sekvencie žmurkania. Správca systému môže kedykoľvek nakonfigurovať nové gestá.

2.4 Všeobecné obmedzenia

- Nepretržitý beh kamery
- Fullscreen živý obraz
- Zanecháva aspect ratio kamery
- Upozornenie pri nedostatku svetla
- Minimálna frekvencia 26 fps
- Windows aplikácia
- Nezávislé od konkrétneho hardvéru
- Nelimitovaný počet gest

2.5 Predpoklady a závislosti

- Vhodná vzdialenosť tváre od kamery
- Pri detekcii viacerých tvárí zvoliť, na ktorú sa sústreďuje
- Umiestnenie v dobre osvetlenom priestore
- Použitie kamery, ktorá má vhodné zorné pole

3 Špecifikácia požiadaviek

3.1 Funkčné požiadavky

3.1.1 Region of interest

Systém bude zaznamenávať celú oblasť kamery, snaží sa nájsť tvár dostatočnej veľkosti na spoľahlivú detekciu žmurkania.

Výber záujmovej časti obrazu, ktorá bude ďalej spracovávaná v riešení bude možné definovať parametrami: dĺžka, šírka, offset x, offset y. Pričom dĺžka a šírka sú definované v pixeloch a určujú veľkosť záujmového okna. Offset X a Offset Y sa zadávajú v pixeloch a definujú posun záujmového okna z bodu 0,0 (ľavý horný roh).

3.1.2 Výber tváre

Aplikácia bude schopná detekovať viacero objektov (tvárí), ale pracovať bude iba s jednou z nich. Aplikácia vyberie tú, čo je bližšie ku stredu a rámikom ju zvýrazní. V konfigurácii bude možné meniť minimálnu požadovanú veľkosť tváre, ak to systém nebude vedieť robiť automaticky.

3.1.3 Zobrazenie obrazu z webovej kamery v reálnom čase

Aplikácia bude na celej obrazovke zobrazovať živý záznam z webkamery. Zároveň aplikácia zachová pôvodný pomer strán (aspect ratio). Ak sa aspect ratio kamery a obrazovky nezhoduje, zvyšok obrazu bude vyplnený čiernymi rámami. Bude responzívna vzhľadom na rozlíšenie displeja.

3.1.4 Grafické zvýraznenie detekovaného objektu

Pri detekcii tváre sa na obrazovke v oblasti tváre zobrazí tenký rámik. Používateľ tak bude informovaný, že sa aplikácia sústreďuje konkrétne na jeho tvár.

3.1.5 Detekcia gest

Systém bude vedieť detekovať nasledujúce úkony:

- U1 Zatvorenie oboch očí
- U2 Zatvorenie ľavého oka
- U3 Zatvorenie pravého oka
- U4 Pohľad doľava
- U5 Pohľad doprava
- U6 Úsmev
- U7 Vyplazený jazyk

Úkony U4-U7 sú voliteľné, budú implementované len ak na to tím bude mať prostriedky.

Každý úkon má nastaviteľnú dobu trvania ako krátky a dlhý (ak to bude vhodné, tak bude na výber z dvoch dlhých). Dĺžka týchto trvaní bude globálne nastaviteľná v konfiguračnom súbore.

Úkony U1-U3 budú oddelené otvorením očí; U4 a U5 pohľadom na displej/kameru; U6 a U7 zatvorenými ústami v neutrálnej pozícii.

3.1.6 Výpis po detekcii gesta

Systém po vykonaní nejakého z gest definovaných v konfiguračnom súbore zobrazí informáciu o tom, že dané gesto bolo vykonané (vykreslením príslušného obrázka alebo textu k danému gestu v rohu okna aplikácie po dobu niekoľkých sekúnd).

3.1.7 Nápoveda gest

Systém bude disponovať nápovedou, ktorá bude obsahovať postupnosti úkonov, ktoré systém dokáže detekovať. Nápoveda sa bude nachádzať na obrazovke po celú dobu behu aplikácie, na spodnej časti obrazu tak, aby nezaberala jeho veľkú časť. Gestá budú mať názov a popis/obrázky, ako ich vykonať.

3.1.8 Konfigurácia gest

Gestá budú definované v konfiguračnom súbore. Správca bude mať umožnené tieto gestá meniť, mazať a pridávať. Gesto sa nastavuje ako postupnosť úkonov definovaných v 3.1.5 a mena súboru s obrázkom, ktorý sa má zobrazit' po jeho detekcii. Časové trvanie zobrazenia detekovaného gesta v milisekundách (pre všetky gestá spoločný) sa tiež bude nastavovať v konfiguračnom súbore.

3.1.9 Ukončenie gesta

Je možné nastaviť časový interval, po ktorého uplynutí sa gesto považuje za ukončené. Až potom sa gesto vyhodnotí, aby sa predišlo zlej detekcii pri gestách obsahujúcich rovnakú začiatočnú postupnosť úkonov.

3.1.10 Ochrana displeja (screen saver)

(Bude implementovaný len ak na to tím bude mať prostriedky). Aplikácia bude fungovať aj v prípade, že displej bude softvérovo vypnutý (uspatý). Po detekovaní tváre sa displej zobudí a na displeji sa zobrazí okno aplikácie. Ak systém nedetekuje žiadnu tvár po nejakú dobu, operačný systém displej uspí.

3.1.11 Varovanie pri nízkom osvetlení

V prípade ak je v miestnosti nedostatok svetla tak bude o tom používateľ informovaný. Nedostatok svetla nastane vtedy, ak je celková priemerná svietivosť nižšia ako je definovaná hodnota (hodnota je definovaná v kóde).

3.2 Požiadavky rozhrania

3.2.1 Hardvérové rozhrania

Systém bude fungovať s ľubovoľnou kamerou, zabudovanou alebo externou, ktorá je použiteľná na operačnom systéme Windows a podporuje API DirectShow.

3.2.2 Možnosť vybrať kameru zo zoznamu kamier

V prípade ak je k počítaču pripojených viac kamier tak je možné vybrať konkrétnu kameru zadáním čísla (poradia) kamery do konfiguračného súboru (predvolená bude kamera s označením 0).

3.2.3 Softvérové rozhrania

Windows aplikácia, nebude (nemusí byť) funkčná, respektíve spustiteľná na zariadeniach s iným operačným systémom.

3.2.4 Užívateľské rozhrania

Užívateľské prostredie bude tvoriť jedno full screen okno, na ktorom budú umiestnené všetky funkcionality pre používateľa.

3.3 Požiadavky, ktoré sa nevzťahujú na funkcionality

3.3.1 Implementačné požiadavky

Aplikácia bude vytvorená na operačnom systéme Windows. Bude vyvíjaná v programovacom jazyku Python a pomocou open-source knižnice na OpenCV, ktorá je určená na prácu s počítačovým videním.

3.3.2 Stručná dokumentácia

K aplikácii bude priložená krátka dokumentácia, ktorá bude obsahovať používateľskú príručku, návod na nastavovanie gest. Priložený bude aj krátky text k exponátu, vhodný aj pre deti.

Návrh

1 Úvod

1.1 Účel dokumentu

Tento dokument slúži ako návrh informačného systému exponátu do Zážitkového centra vedy Aurelium. Dokument dôkladne popisuje funkcie a metódy informačného systému pomocou diagramov a popisov, spôsob ako bude systém vyvinutý a ako bude následne fungovať tak, aby spĺňal všetky požiadavky obsiahnuté v Katalógu požiadaviek.

2 Dekompozícia a popis

2.1 Python

Python je interpretačný programovací jazyk. Využíva objektovo-orientovaný prístup. Kód aplikácie bude napísaný v tomto jazyku. Budeme využívať najnovšiu verziu tohto programovacieho jazyka (3.7).

2.1.1 OpenCV

OpenCV je knižnica pre python na spracovanie a zaznamenávanie obrazu. Zároveň daná knižnica vie komunikovať s vonkajšími snímacími zariadeniami (v našom prípade ide o webkameru). Po nasnímaní obrazu sú jednotlivé snímky prevedené z RGB kódovania na čiernobiele kódovanie. Okrem toho nám táto knižnica dovoľuje pracovať s oknom aplikácie a umožňuje kreslenie jednoduchých tvarov a obrázkov, ktoré využijeme na označenie detekovanej časti.

2.1.1.1 OpenCV Funkcie

- `cv2.VideoCapture(int:numberOfCamera(starting with 0))` - sníma video z príslušnej kamery
- `cv2.putText(frame:frame, string:text)` - vypíše text
- `cv2.rectangle(frame:frame, tuple:coords, tuple:coords, color:color, int:width)` - nakreslí obdĺžnik na dané miesto
- `cv2.imshow(string:name, frame:frame)` - vykreslí daný obrázok
- `cv2.waitKey(int:number)==int:number` - čaká pokiaľ nie je zmačknutý kláves
- `cv2.destroyAllWindows()` - zahodí vytvorené okná
- `cv2.resize(frame:frame, tuple:size)` - mení veľkosť obrázka
- `cv2.cvtColor(frame:frame, color:color)` - vyberá farebné kódovanie (`cv2.COLOR_BGR2GRAY`)
- `cv2.line(frame:frame, tuple:tuple, tuple:tuple, color:color)` - nakreslí čiaru
- `cv2.bilateralFilter(frame:frame, int:distance)` - aplikuje bilateral filter
- `cv2.erode(frame:frame, algorithm:algo, int:iterations)` - eroduje obraz použitím štruktúrneho elementu
- `cv2.threshold(frame:frame, int:thresholdWhite, int:thresholdBlack, cv2.THRESH_BINARY)` - zmení farby obrazu na čiernu a bielu podľa obmedzenia
- `cv2.countNonZero(frame:frame)` - počet nonZero framov
- `cv2.fillPoly(frame:frame, tuple:points, color:color)` - vyplní polygón danou farbou
- `cv2.bitwise_not(frame:frame, frame:frame, mask:mask)` - prevráti všetky bity v zozname
- `cv2.moments(mask:mask)` - pomáha rozpoznať určité charakteristiky obrazu
- `cv2.findContours(frame:frame, cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)` - nájde všetky kontúry objektu
- `cv2.namedWindow(frame:frame, property:prop)` - vytvorí okno s danou vlastnosťou (`cv2.WND_PROP_FULLSCREEN`)
- `cv2.setWindowProperty(frame:frame, property:prop, property:prop, ...)` - pridá vlastnosť existujúcemu oknu
- `camera.read()` - vráti nový frame kamery
- `camera.get(property:prop)` - ukáže fps alebo iné vlastnosti
- `camera.isOpened()` - zistí, či je kamera aktívna

2.1.2 NumPy

Knižnica pre Python určená na prácu s viacrozmernými poľami a maticami.

2.1.2.1 NumPy Funkcie

- `numpy.ndarray(shape:shape, int:offset)` - multidimenzionálny zoznam prvkov fixnej veľkosti
- `numpy.ones(shape:shape, int:order)` - vytvorí zoznam daného tvaru a typu, vyplnený jednotkami
- `numpy.zeros((int:height, int:width), dtype:numpy.uint8)` - vytvorí zoznam daného tvaru a typu, vyplnený nulami
- `numpy.array(object:obj, dtype:type)` - vytvorí zoznam
- `numpy.max(region:fromto)` - maximum zoznamu elementov
- `numpy.min(region:fromto)` - minimum zoznamu elementov
- `np.full((int:height, int:width), int:value, dtype:type)` - vytvorí zoznam daného tvaru a typu, vyplnený danou hodnotou

2.1.3 GazeTracking

Knižnica pre Python, ktorá obsahuje naučený algoritmus pre snímanie tváre, očí a pohľadu doľava/doprava.

Modul containing pre-trained eye detection algorithm

2.2 CMake

Open-source software použitý na vybudovanie projektu z C++ do Python. Pre správne fungovanie sa vyžaduje C++ kompilátor vo vhodnej verzii (32/64 bit). U nášho projektu to slúži na nainštalovanie C++ modulov do Pythonu.

3 Návrh používateľského rozhrania

3.1 UI

Aplikácia bude zaberat' celú obrazovku. Pokiaľ sa nezhoduje aspect ratio kamery a obrazovky, tak sa obrazovka doplní čiernymi barmi. V celom okne bude zobrazený živý obraz kamery.

Po detekovaní vhodnej tváre aplikácia túto tvár orámuje jemným rámikom. Týmto dá vedieť používateľovi, že ho kamera správne detekuje.

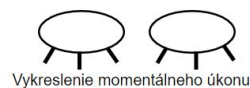
V pravom hornom rohu bude zobrazený momentálny úkon, ktorý používateľ vykonáva.

V ľavom hornom rohu sa na pár sekúnd po úspešnom vykonaní gesta zobrazí textový výpis nastaviteľný v konfiguračnom súbore.

Na spodku aplikácie bude nápoveda gest. Bude pokrývať záznam kamery, jej plnej šírky, polopriehľadným obdĺžnikom. Gestá sa na nej budú zobrazovať z ľava do prava. Pre každé nakonfigurované gesto bude zobrazené jeho meno a obrázky znázorňujúce postupnosť úkonov potrebnú na správne vykonanie gesta.

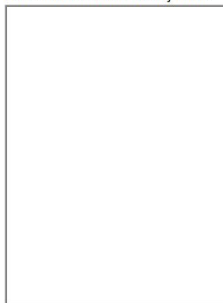


Výpis/obrázok/animácia
po rozpoznaní gesta



Vykreslenie momentálneho úkonu

Orámovanie sledovanej tváre



Nápoveda gest



4 Návrh implementácie

4.1 UML - Use-case diagram

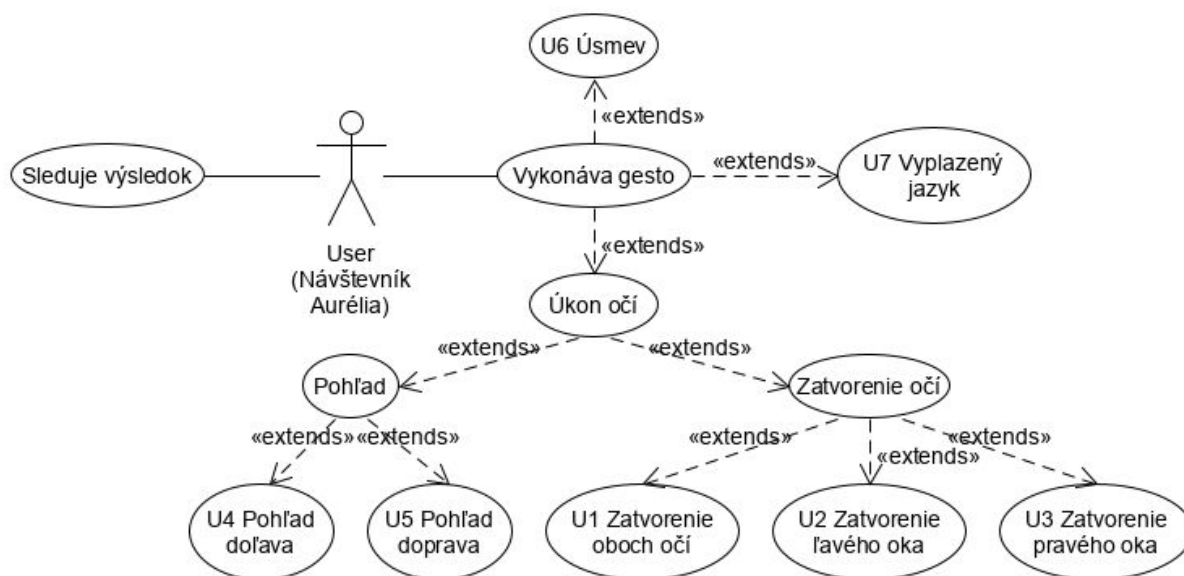


Diagram zobrazuje činnosti návštevníka Aurélia, ktoré program rozpoznáva.

4.2 UML - Stavový diagram

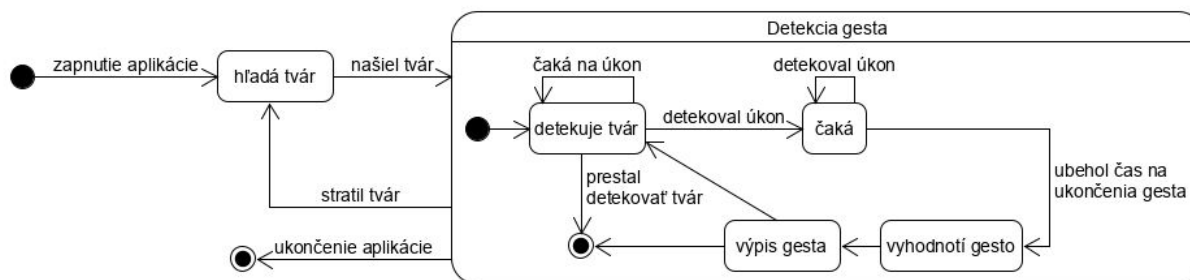
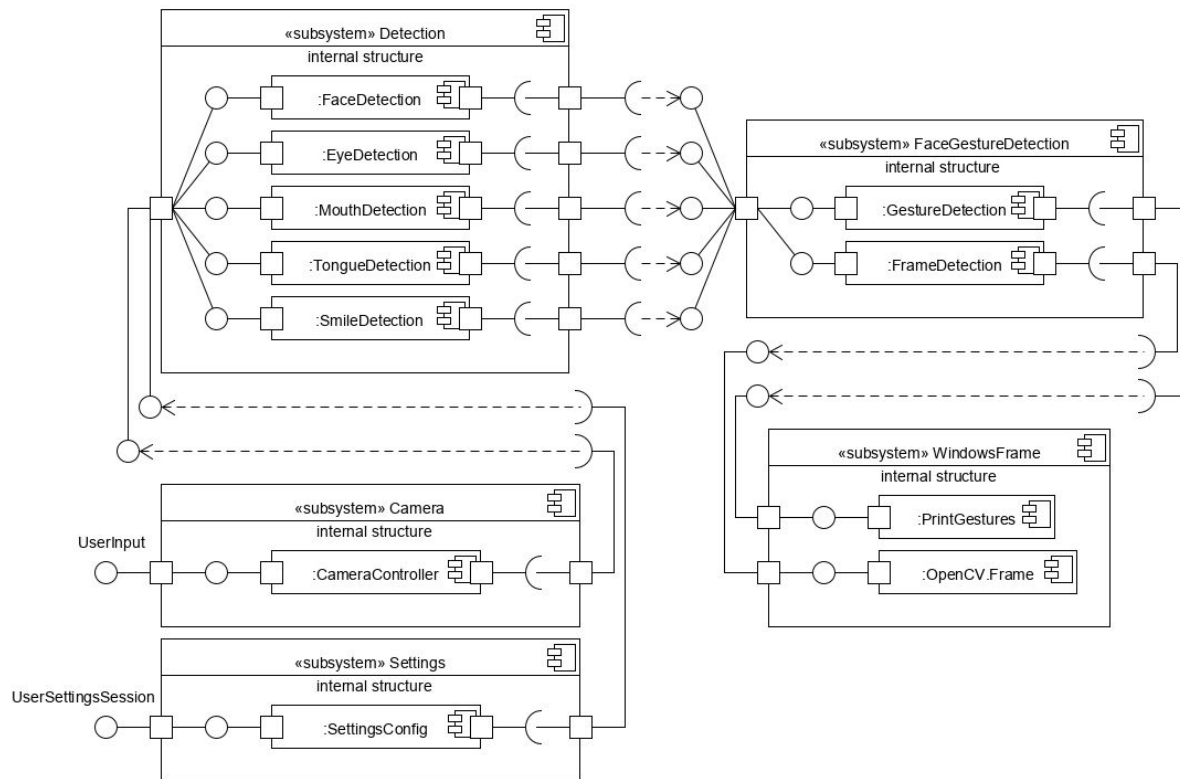
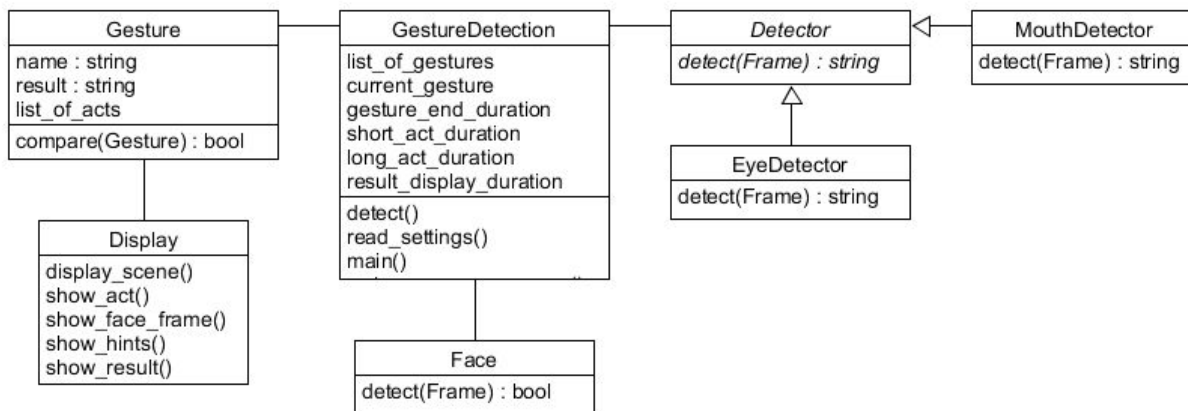


Diagram zobrazuje možné stavy programu.

4.3 UML - Diagram komponentov



4.4 UML - Triedny diagram



Gesture - gesto

name - meno daného gesta

result - text priradený ku gestu, ktorý bude zobrazený po jeho správnom vykonaní

list_of_acts - postupnosť úkonov potrebných na vykonanie gesta

compare(Gesture) - porovná dve gestá a zistí, či sa zhodujú

Display - trieda obsahujúca funkcie, ktoré zobrazia elementy na obrazovke

display_scene() - na celej obrazovke bude bežať živý obraz z kamery

show_act() - v pravom hornom rohu zobrazí momentálne vykonávaný úkon

show_face_frame() - tenkým rámikom orámuje detekovanú tvár

show_hints() - na spodku obrazovky zobrazí nápovedu ako vykonať prednastavené gestá

show_result() - v ľavom hornom rohu vypíše Gesture result

GestureDetection

list_of_gestures - zoznam všetkých prednastavených gest, ktoré aplikácia rozpoznáva

current_gesture - postupnosť momentálne vykonaných úkonov

gesture_end_duration - čas čakania po poslednom úkone na ukončenie gesta

short_act_duration - čas trvania krátkeho úkonu

long_act_duration - čas trvania dlhého úkonu

result_display_duration - dĺžka zobrazenia výpisu po vykonaní gesta

detect() - zistí, či sa zadané gesto nachádza v zozname rozpoznateľných gest

read_settings() - prečíta a nastaví nastavenia z konfiguračného súboru

main() - tu bude bežať celý program

wake_up_screen_saver() - vypne screen saver

Face - trieda na detekciu tváre

`detect (Frame)` - vráti, či momentálne detekuje tvár

Detector - abstraktná trieda

`detect (Frame)` - vráti meno rozpoznaného úkonu

EyeDetector - detekuje úkony očí

`detect (Frame)` - upravená funkcia, ktorá zistí, ktorý úkon očí bol vykonaný

MouthDetector - detekuje úkony úst

`detect (Frame)` - upravená funkcia, ktorá zistí, ktorý úkon úst bol vykonaný

4.5 Problém kalibrácie

Prvou hlavnou úlohou pri vytváraní aplikácie, ktorá detekuje žmurkanie používateľov bolo vybrať správny detektor na snímanie žmurkania. Našou podmienkou bolo to, aby išlo o detektor naprogramovaný v programovacom jazyku Python a využíval knižnicu OpenCV, využívanú hlavne na počítačové videnie.

Na výber bolo viacero takýchto detektorov.

Jeden z prvých, ktorý sme skúšali je Haar cascade. Ide o starý detektor, ktorý je ale veľmi rýchly a ľahko implementovateľný. Tento detektor sme ale nepoužili, pretože presnosť detekovania tváre bez učenia bola veľmi nízka.

Následne sme skúšali aplikácie využívajúce face landmarks. Tieto už fungovali lepšie, avšak sme neboli stále spokojní s presnosťou. Okrem face landmarkov využíva tzv. hodnotu EAR (eyes aspect ratio). Je to vlastne výpočet medzi vertikálnou a horizontálnou hodnotou oka. Cez tento výpočet vieme zistiť, či je oko zavreté alebo nie. Podľa tohto netreba spracovávať obraz detailne a výpočet je veľmi rýchly a efektívny. Avšak účinnosť ani tohto softvéru nebola dobrá.

Nakoniec sme využili podobný systém (ktorý tiež využíval face landmarks a počítal hodnotu EAR), avšak mal v sebe ešte kalibráciu na zlepšenie presnosti. Okrem toho sme vytvorili ešte jednu internú kalibráciu, ktorá neberie ako referenciu každý snímok, ale podľa hodnoty premennej (ktorá je meniteľná), sa najprv určí, koľko snímok spracuje predtým než zobrazí výsledok na obrazovku. Čím je toto číslo väčšie, tým je výsledok presnejší, avšak output potom trvá dlhšie. Momentálne sme sa usadili s číslom 3, ktoré je dostatočne rýchle a zároveň má slušnú presnosť.

Posledným typom systému, ktorý sme ešte skúšali bol Keras a Tenserflow. Táto aplikácia využíva hlavne GPU značky NVIDIA (CUDA jadra) na výpočty. Bez takého jadra bola detekcia veľmi pomalá. Preto sme sa tomuto systému vyhli.

5 Špecifikácia vonkajších interfejsov

Aplikácia bude komunikovať a prijímať vstupy z kamery. Môže ísť o internú kameru alebo kameru pripojenú cez USB. Komunikácia bude zabezpečená prostredníctvom knižnice openCV, ktorá pomocou modulu cv2 bude čítať dáta z kamery.

6 Formáty súborov

Systém bude implementovaný v jazyku Python, teda implementačné súbory budú vo formáte .py. Systém bude obsahovať jeden konfiguračný súbor vo formáte .txt na popisovanie gest.

6.1 Konfiguračný súbor

Štruktúra config súboru bude nasledovná: jeden riadok znamená jedno nastavenie. Po spracovaní config súboru si to v programovacom jazyku uložíme do mapy, z ktorej budeme pristupovať k jednotlivým elementom z nastavení. Gestá budú uložené v triede Gesture, kde bude aj zoznam obsahujúci celistvé postupnosti gest.

Idea mapy výpisov je nasledovná (gests predstavuje mapu) : `gests["LRlrmc"] = "Vypis popis gesta"`

- < - pohľad doľava
- > - pohľad doprava
- l - zatvorené ľavé oko krátko
- r - zatvorené pravé oko krátko
- L - zatvorené ľavé oko dlho
- R - zatvorené pravé oko dlho
- M - otvorené ústa
- S - úsmev

Riadky súboru budú definované nasledovne:

- Prvý riadok bude číslo kamery
- Druhý riadok bude určovať dĺžku pauzy v ms, po ktorej sa vyhodnotí gesto
- Tretí riadok bude určovať dĺžku krátkeho úkonu v ms
- Štvrtý riadok bude určovať dĺžku dlhého v ms
- Piaty riadok bude určovať dĺžku zobrazenia výpisu po rozpoznaní gesta
- Každý ďalší riadok bude definovať gesto
 - najprv bude názov gesta v úvodzovkách
 - po medzere výpis po rozpoznaní v úvodzovkách
 - po ďalšej medzere bude postupnosť úkonov označených znakom "názov gesta" "výpis" l<MR

7 rozdelenie práce

- UI - Martina
- pomer strán, čierne bary - Martina
- konfiguračný súbor a jeho čítanie - Ivana
- kalibrácia dĺžky žmurkania - Ivana, Martina
- kalibrácia detekcie - Miroslav
- Zameranie sa na jednu osobu - Matej
- kontrola vykonania gesta - Matej
- kontrola dostatku svetla - Miroslav
- testovanie programu s rôznymi kamerami - všetci

Testovacie scenáre

1 Oblasť záujmu

1.1 Návštevník je príďaleko / nie je pri exponáte

Akcia: Na výstavisku sa prechádzajú návštevníci, ale žiaden z nich nie je v dostatočnej blízkosti kamery.

Reakcia: Systém nedetekuje žiadnu tvár.

1.2 Návštevník nie je v zornom poli detekcie

Akcia: Návštevník podíde bližšie k exponátu, ale je tak na kraji, že mu kamera takmer nezaznamenáva celú tvár.

Reakcia: Systém nedetekuje žiadnu tvár.

1.3 Návštevníkova tvár je na vhodnom mieste

Akcia: Návštevník podíde kúsok pred kameru a je takmer v strede displeja.

Reakcia: Systém detekuje tvár, zobrazí okolo nej rámik.

2 Zobrazenie obrazu

2.1 Kamera sníma

Akcia: Kamera beží a návštevníci sa hrajú na výstavisku.

Reakcia: Na displeji sa zobrazuje živý obraz z kamery v reálnom čase a taktiež všetká funkcionálna aplikácia.

2.2 Širokoúhlý monitor ale nie kamera

Akcia: Na výstavisku majú veľmi širokoúhlý monitor, ale ich kamera nie je schopná zamerať také rozlíšenie.

Reakcia: Po boku obrazu sa zobrazí čierne orámovanie takej šírky, aby nebol obraz zdeformovaný.

2.3 Širokohlá kamera ale nie monitor

Akcia: Používaná kamera je širokohlá ale monitor má menšie aspect ratio.

Reakcia: Na vrchu a spodku obrazovky bude čierny rám, ktorý dopĺňa monitor o rozdiel v aspect ratio.

2.4 Nedostatok svetla pri kamere

Akcia: Kamera nedokáže dobre detekovať tvár kvôli nedostatku svetla v miestnosti.

Reakcia: Systém bude vypisovať chybu o nedostatku svetla v miestnosti pokiaľ sa nezlepší osvetlenie v miestnosti (inak nebude možné snímať tvár).

3 Zvýraznenie detekovaného objektu a ich počet

3.1 Detekovaná tvár

Akcia: Návštevník príde k exponátu a systém detekuje jeho tvár.

Reakcia: V oblasti okolo jeho tváre sa vykreslí rámik, aby mu bolo jasné že ho systém detekoval.

3.2 Viacero Tvárí

Akcia: Okolo exponátu sa zhromaždí viacero ľudí.

Reakcia: Systém detekuje tvár čo je v oblasti záujmu a vhodnej veľkosti na detekciu. Vykreslí okolo nej rámik, pre ostatných pozorujúcich sa rámik nevykreslí.

4 Detekcia gest

4.1 Správne vykonanie gesta

Akcia: Návštevník žmurká gesto z nápovedy, nachádzajúcej sa na spodku obrazovky a po dokončení počká chvíľku na výsledok.

Reakcia: Vypíše sa výsledok gesta v ľavom hornom rohu obrazovky. Text výsledku a doba na ktorú sa má zobrazit' je nastavená v konfiguračnom súbore.

4.2 Zlé vykonanie gesta

Akcia: Návštevník žmurká gesto z nápovedy, ale pomýli sa pri ňom pretože jeden úkon z neho vykoná prikrátka.

Reakcia: Na obrazovke sa nezobrazí výpis po vykonaní gesta

4.3 Prerušenie vykonania gesta iným objektom

Akcia: Návštevník žmurká gesto z nápovedy, ale do cesty príde iný objekt (človek alebo vec).

Reakcia: Pokiaľ bola dĺžka na zaznamenanie gesta dostatočná, tak sa urobí záver po dobu, pokiaľ bol návštevník snímaný.

5 Konfigurácia a zapnutie

5.1 Zmena nastavení

Akcia: Správca exponátu sa rozhodne zmeniť gestá alebo časy, otvorí konfiguračný súbor a podľa priloženého predpisu gest ich zmení.

Reakcia: Po spustení aplikácie systém načíta z konfiguračného súboru časy a gestá a bude bezproblémovo fungovať s aktualizovanými údajmi.

5.2 Nevhodný konfiguračný súbor

Akcia: Konfiguračný súbor obsahuje chybné napísané informácie.

Reakcia: Po spustení aplikácie sa zobrazí error message, ktorý oboznámi používateľa, že konfiguračný súbor je chybný.

6 Ochrana displeja

6.1 Nič sa nedeje

Akcia: Na výstavisku sa dlho nič nedeje, nie sú tam žiadne pohyby.

Reakcia: Obrazovka sa uspí.

6.2 K exponátu sa priblíži návštevník

Akcia: Obrazovka je v móde spánku a k exponátu sa priblíži návštevník.

Reakcia: Obrazovka sa zapne a pokračuje v behu aplikácie.

6.3 Kamera neustále sníma návštevníka

Akcia: Obrazovka je zapnutá a sníma návštevníka.

Reakcia: Ochrana displeja sa nezapne pokým aplikácia detekuje tvár.

7 Kamera

7.1 Viacero kamier

Akcia: K počítaču sú pripojené viaceré kamery.

Reakcia: Systém vyberie tú, ktorá je určená v konfiguračnom súbore.

7.2 Žiadna kamera

Akcia: Počítač nemá k dispozícii žiadnu kameru.

Reakcia: Aplikácia sa vypne a vypíše výnimku - dôvod nefunkčnosti.

8 Detekcia úkonov

8.1 Zatvorenie oboch očí

Akcia: Návštevník zatvorí obe oči.

Reakcia: V pravom hornom rohu obrazovky sa zobrazí obrázok zatvorených očí až kým nezačne vykonávať ďalší úkon.

8.2 Zatvorenie ľavého oka

Akcia: Návštevník zatvorí ľavé oko.

Reakcia: V pravom hornom rohu obrazovky sa zobrazí obrázok zatvoreného ľavého oka a otvoreného pravého oka až kým nezačne vykonávať ďalší úkon.

8.3 Zatvorenie pravého oka

Akcia: Návštevník zatvorí pravé oko.

Reakcia: V pravom hornom rohu obrazovky sa zobrazí obrázok otvoreného ľavého oka a zatvoreného pravého oka až kým nezačne vykonávať ďalší úkon.

8.4 Pohľad doprava

Akcia: Návštevník sa pozrie do prava.

Reakcia: V pravom hornom rohu obrazovky sa zobrazí obrázok očí pozerajúcich sa doprava až kým nezačne vykonávať ďalší úkon.

8.5 Pohľad doľava

Akcia: Návštevník sa pozrie doľava.

Reakcia: V pravom hornom rohu obrazovky sa zobrazí obrázok očí pozerajúcich sa doľava až kým nezačne vykonávať ďalší úkon.

8.6 Úsmev

Akcia: Návštevník sa usmeje.

Reakcia: V pravom hornom rohu obrazovky sa zobrazí obrázok úsmevu až kým nezačne vykonávať ďalší úkon.

8.7 Vyplazený jazyk

Akcia: Návštevník vyplazí jazyk.

Reakcia: V pravom hornom rohu obrazovky sa zobrazí obrázok jazyka až kým nezačne vykonávať ďalší úkon.

Návod

Ak chce prevádzkovateľ exponátu zmeniť gestá, ich výpisy alebo dĺžky časov, je potrebné prepísať súbor *configuration.txt*.

Pri zmene je potrebné dodržať štruktúru, a to tak aby:

1. Prvý riadok bolo číslo kamery
2. Druhý riadok je číslo, ktoré určuje dĺžku pauzy v ms, po ktorej sa vyhodnotí gesto
3. Tretí riadok je číslo, ktoré určuje dĺžku krátkeho úkonu v ms
4. Štvrtý riadok je číslo, ktoré určuje dĺžku dlhého úkonu v ms
5. Piaty riadok je číslo, ktoré určuje dĺžku zobrazenia výpisu po rozpoznaní gesta
6. Všetky nasledujúce riadky definujú gestá
 - štruktúra riadku: “názov gesta”“výpis”postupnosť úkonov
 - príklad: “pozdrav”“Ahoj”Lrb (znamená že po postupnosti troch gest dlhé žmurknutie ľavým okom, krátke pravým a krátke oboma sa vypíše na obrazovke Ahoj)
 - Úkony, z ktorých sa gesto môže skladať:
 - § L = dlhé zavretie ľavého oka
 - § R = dlhé zavretie pravého oka
 - § B = dlhé zavretie oboch očí súčasne
 - § l = krátke zavretie ľavého oka
 - § r = krátke zavretie pravého oka
 - § b = krátke zavretie oboch očí súčasne